Feature extraction and performance measure of Requirement Engineering (RE) document using text classification technique

DR. L P SAIKIA (PROFESSOR)

Department of computer science & Engineering

Assam Downtown University

Panikhaiti – Guwahati (Assam)

Email: lp_saikia@yahoo.co.in

SHILPI SINGH (RESEARCH SCHOLAR)

Department of computer science & Engineering

Assam Downtown University

Panikhaiti – Guwahati (Assam)

Email: shilpi24_singh@yahoo.co.in

***Abstract*— The RE document in the SDLC phase of software development is prone to ambiguity, since it is written in natural language. The text classification is a method of assigning a document as predefined classes or categories. The efficient understanding of text document is important to improve the quality of RE document, and this can be achieved by using semantic information regarding a text document. The main objective of this experimentation is to utilize semantic information to identify features and prepare data sets for better classification of text as "Ambiguous" or "Unambiguous". The different data sets are constructed and then are analyzed both manually as well as computationally on different parameters (kappa index and likelihood ratio) to understand the quality of any of RE documents.**

Keywords*— *RE document (Requirement engineering document); kappa index; POS tagging; Likelihood ratio; Decision tree based C4.5 algorithm.

I. INTRODUCTION

The Requirement elicitation is the important phase of software development in which the exact requirement of the system is specified. The exact system definition or specification is important before building any system. But RE document is always written in natural language English, and is susceptible to ambiguity. Thus it is very important to measure, detect and identify any text ambiguity in RE document to decrease the cost and time of software development [1]. The main objective of requirement analysis and specification phase is to clearly understand the user's requirement and to systematically arrange the requirement into a specification document called as SRS (Software requirement specification) document that are mostly documented using natural language. The RE document is unambiguous if every stated requirement has only one meaning. Since, the stakeholders are from different background thus they may misunderstand the stated requirement. Natural language is the commonly used scheme or technique for expressing the exact requirements in industry and thus can be ambiguous if not understand properly. The majority of requirement documents are written in natural language as the survey by shows [2]. The customers and software developers if not agree conceptually and then they may lead to miss the deadline of software and the cost of development also increases. In software development it is always good to detect any kind of textual ambiguity as early as possible to avoid software crisis or failure. A mathematical and statistical study [3] indicates that in majority of the cases the text are easily misinterpreted and thus need to be identified in the early stages of software development. And thus it is always good to use formal methods for specifying a RE document, since it is mathematical model for specifying RE text. Ambiguity means that one word or sentence can be interpreted in several ways. Ambiguity in the code or in document especially in the case of safe critical systems may have disastrous results. No less serious problems caused by ambiguous description of requirements, which besides influencing the success of the project in technical terms may also have damaging consequences in contractual terms [4]. The RE document consists of different types of ambiguity that may vary from lexical ambiguity to semantic ambiguity and pragmatic ambiguity.

II. STAGES OF TEXT CLASSIFICATION

The main stages in text classification technique are:

a) Document Selection

b) Document Preprocessing

c) Feature Selection

d) Data Representation using *.arff* format.

e) Performance Evaluation using WEKA

978-1-5386-3039-6/18/$31.00©2018 IEEE